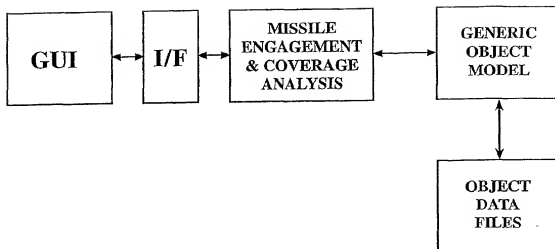
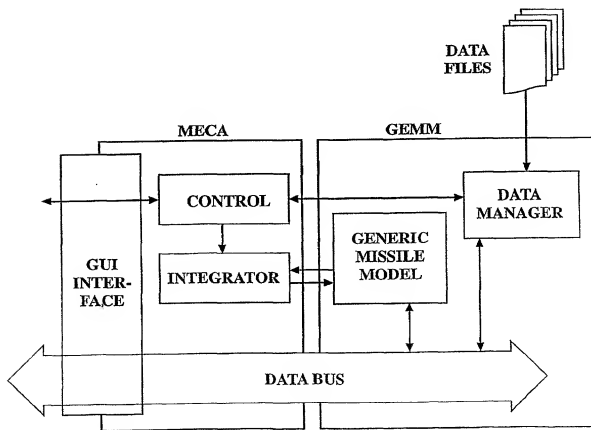


1/10

**Fig 1****FIG 2**

2/10

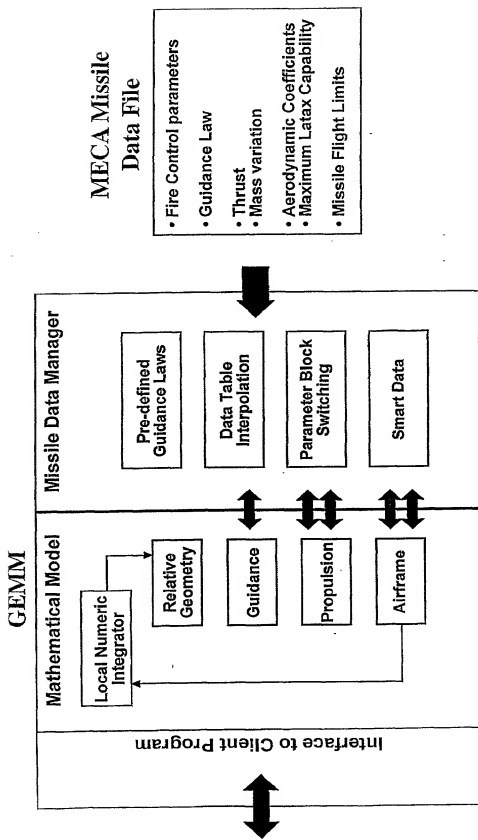
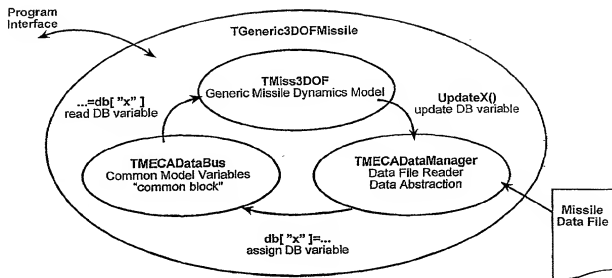


Fig 3

3 / 10

**Fig 4**

10027005-122001

4/10

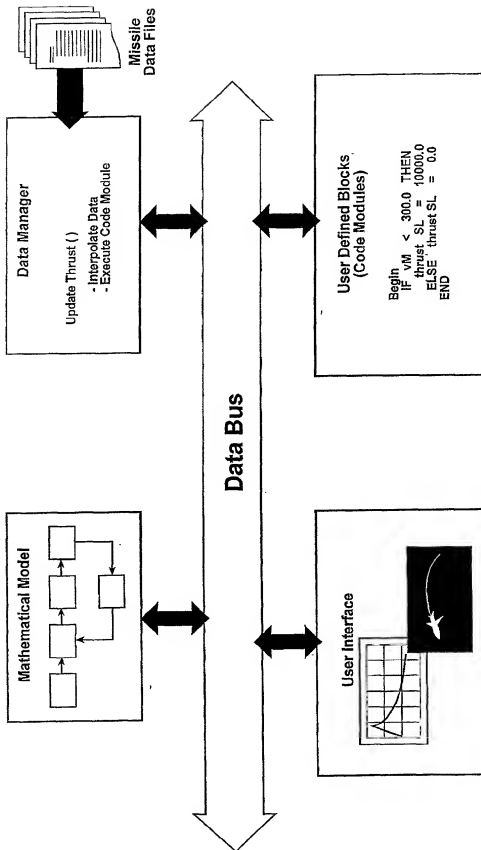


Fig 5

5/10

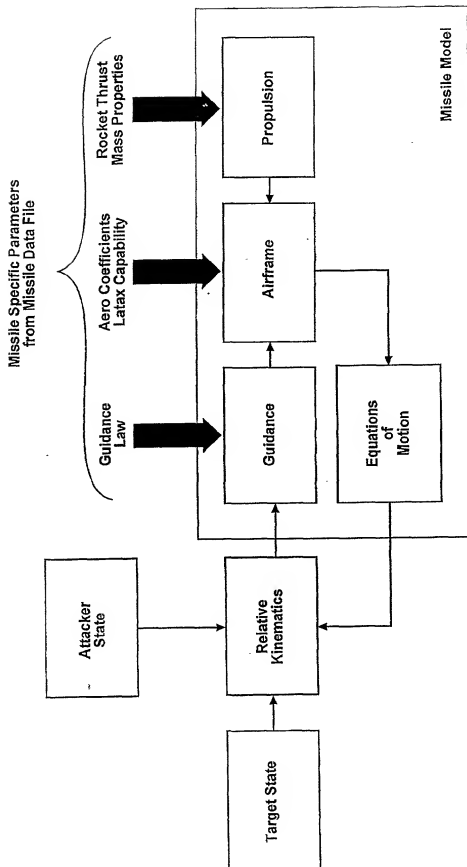


Fig 6

## Execution Process:

1. Data manager loads missile file, parsing it into an internal representation.
2. Data manager updates initial state of data bus variables and internal model options.
3. Model executes for each simulation time-step, calling user defined module where necessary via the StackMac virtual machine.

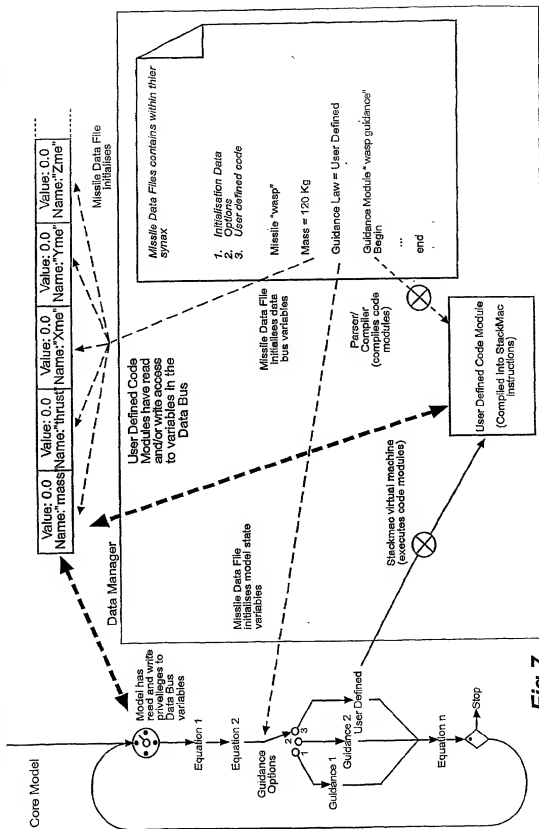


Fig 7

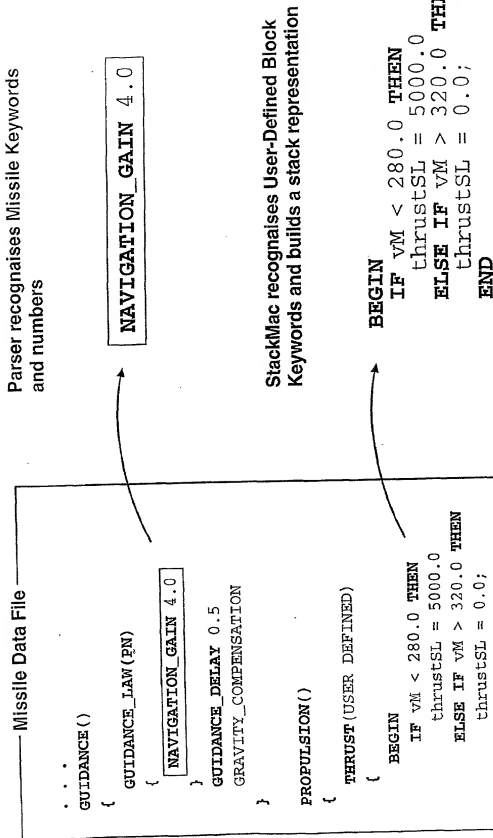


Fig 8

8/10

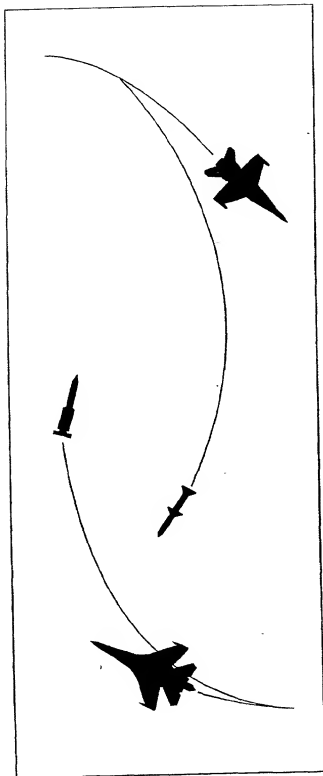


Fig 9

10027625.102001  
1002001.5002001



9/10

### Programstartup

```
// Create Engagement Objects
Engagement model;
...
```

```
// Set Attacker Parameters
model.SetAttackerPosition(x,y,z)
model.SetAttackerSpeed(v);
model.SetAttackerHeading(psi,theta);
...
// Set model time step
model.SetTimeStep(0.1);
...
// Run Engagement model
while (result != FINISHED)
{
    result = model.Step();
}
```

TABLE 1

10/10

## Programstartup

```

// Create Aircraft Objects
TAircraft redAircraft;
TAircraft blueAircraft;

// Create Missile Objects
TMissile redMissile;
TMissile blueMissile;
...

// Load Missile Data Files
redMissile.Load("red.mis")
blueMissile.Load("blue.mis")
...

```

```

// Run Simulation
while (result != FINISHED)
{
    // Propagate aircraft
    redAircraft.Propagate();
    blueAircraft.Propagate();

    // Propagate missiles
    if (redLaunched)
        redMissile.Propagate(blueAircraft);
    if (blueLaunched)
        blueMissile.Propagate(redAircraft);

    // Check simulation STOP logic
    result = Checkstop();
}
...

```

TABLE 2